



## Taller: Análisis de sentimientos

Se presenta el módulo vaderSentiment, una herramienta de análisis de sentimientos que asigna puntuaciones de polaridad a piezas de texto. Se presentará el enfoque basado en reglas y léxicos utilizado por VaderSentiment y su funcionamiento en el análisis de sentimientos.

A continuación, se realizará una demostración práctica del uso de VaderSentiment para analizar el sentimiento de una oración de ejemplo.

```
from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer

# Inicializar el analizador de sentimientos VADER
analyzer = SentimentIntensityAnalyzer()

# Analizar el sentimiento de una oración de ejemplo
sentence = "This movie is fantastic!"
sentiment_score = analyzer.polarity_scores(sentence)

# Imprimir el resultado del análisis de sentimientos
print(sentiment_score)
```



**Output: {'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'compound': 0.5983}**

Utilizarán el conjunto de datos proporcionado y completarán el código para cargar los datos, aplicar el análisis de sentimientos y evaluar el rendimiento del análisis.

```
import pandas as pd
from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
```

```
# Cargar el conjunto de datos de tweets
data = pd.read_csv("twitter_sentiment_data.csv")
```

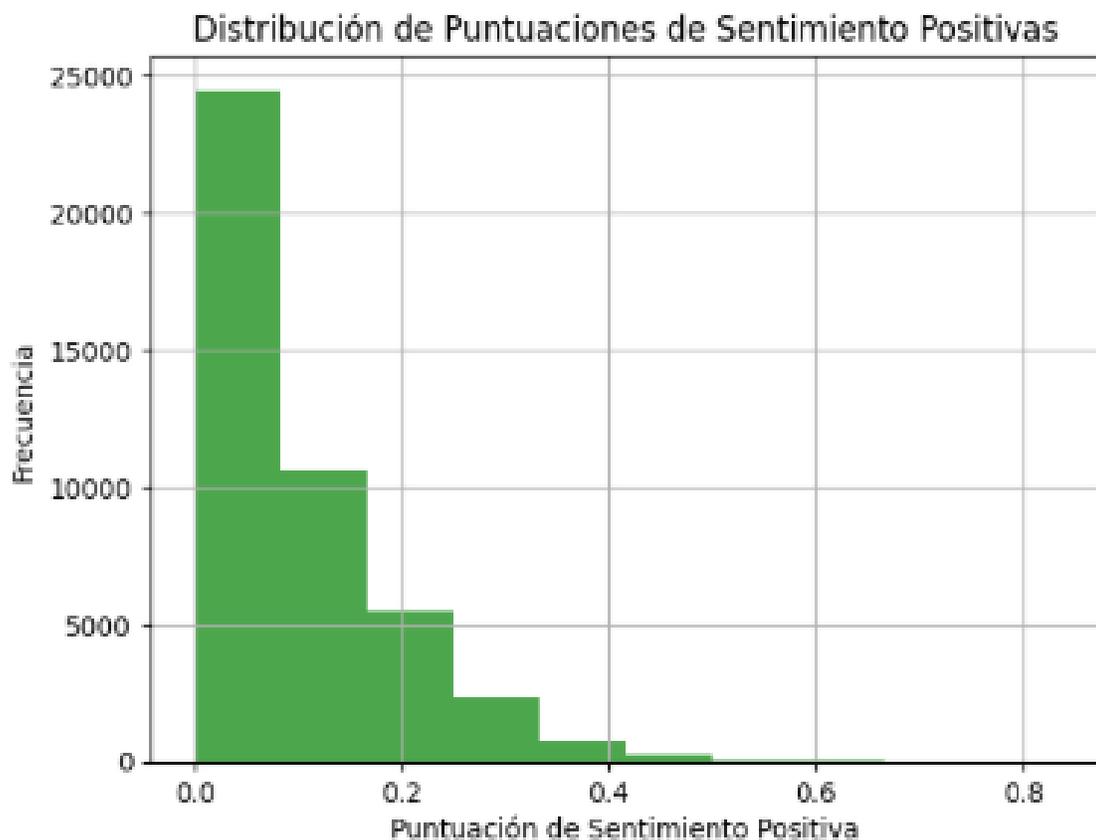
```
# Inicializar el analizador de sentimientos VADER
analyzer = SentimentIntensityAnalyzer()
```

```
# Aplicar el análisis de sentimientos a cada tweet
sentiment_scores = []
for tweet in data['message']:
    sentiment_score = analyzer.polarity_scores(tweet)
    sentiment_scores.append(sentiment_score)
```

```
# Agregar los resultados del análisis al DataFrame
original
data['vader_sentiment'] = sentiment_scores
```

# Imprimir los primeros tweets con sus puntuaciones de sentimiento

```
print(data[['message', 'vader_sentiment']].head())
```



Se solicitará a los participantes que completen el código proporcionado para cargar un nuevo conjunto de datos de tweets, aplicar el análisis de sentimientos utilizando VaderSentiment y luego realizar una comparación de los resultados obtenidos. Consultar [Kaggle](#) para extraer un nuevo conjunto de datos.



```
import pandas as pd
from vaderSentiment.vaderSentiment import
SentimentIntensityAnalyzer
```

```
# Función para cargar los nuevos datos
def cargar_datos(path):
```

```
    """
```

```
    Carga un nuevo conjunto de datos de tweets desde
    el archivo especificado.
```

```
    Args:
```

```
    path (str): Ruta del archivo CSV que contiene los
    datos.
```

```
    Returns:
```

```
    pandas.DataFrame: DataFrame con los nuevos datos
    cargados.
```

```
    """
```

```
    # Completar la carga de datos desde el archivo CSV
```

```
    # data = pd.read_csv(path)
```

```
    # Retornar el DataFrame con los datos cargados
```

```
    # return data
```



*# Función para aplicar el análisis de sentimientos*

```
def analisis_sentimientos(data):
```

```
    """
```

*Aplica el análisis de sentimientos utilizando VaderSentiment a cada tweet en el DataFrame dado.*

*Args:*

*data (pandas.DataFrame): DataFrame que contiene los tweets.*

*Returns:*

*pandas.DataFrame: DataFrame actualizado con los resultados del análisis de sentimientos.*

```
    """
```

```
# Inicializar el analizador de sentimientos VADER
```

```
analyzer = SentimentIntensityAnalyzer()
```

```
# Aplicar el análisis de sentimientos a cada tweet
```

```
sentiment_scores = []
```

```
for tweet in data['message']:
```

```
    sentiment_score = analyzer.polarity_scores(tweet)
```

```
    sentiment_scores.append(sentiment_score)
```

```
# Agregar los resultados del análisis al DataFrame original
```



```
data['vader_sentiment'] = sentiment_scores
```

```
# Retornar el DataFrame actualizado  
return data
```

```
# Función para realizar la comparación de resultados  
def comparar_resultados(data):
```

```
    """
```

Realiza una comparación de los resultados del análisis de sentimientos con los valores de referencia.

Args:

data (pandas.DataFrame): DataFrame que contiene los resultados del análisis de sentimientos.

Returns:

None

```
    """
```

```
# Completar la comparación de resultados con los  
valores de referencia
```

```
pass
```

```
# Cargar los nuevos datos
```

```
# nuevos_datos = cargar_datos("ruta/al/archivo.csv")
```



```
# Aplicar el análisis de sentimientos  
# nuevos_datos =  
analisis_sentimientos(nuevos_datos)
```

```
# Realizar la comparación de resultados  
# comparar_resultados(nuevos_datos)
```